UTF-8 vejledning

Vejledning til indkodning af statistikfiler og tekstfiler som UTF-8. Supplement til brugervejledning til ASTA (Aflevering af Statistikfiler Til Arkiv).

Rigsarkivet august 2020

Version 2.0

Indhold

0. Læsevejledning

Offentlige myndigheder, herunder forskningsinstitutioner, afleverer data til Rigsarkivet i form af arkiveringsversioner og afleveringspakker. Krav til disse afleveringer er beskrevet i Rigsarkivets bekendtgørelse om arkiveringsversioner nr. 128. Et af kravene er, at de data, der afleveres, skal indkodes som UTF-8.

UTF-8 vejledningen er en teknisk vejledning, der forklarer, hvad UTF-8 indkodning er, hvordan du tjekker, om tegnsættet i en tekstfil er indkodet som UTF-8 samt, hvordan du indkoder tegnsæt som UTF-8 i statistikprogrammerne SAS, Stata og SPSS eller i en teksteditor.

A. Vejledningens målgruppe og anvendelse

UTF-8 vejledningen henvender sig til dem, som producerer afleveringspakker med dataudtræk fra statistikfiler til arkivet.

B. Henvisning til øvrig vejledning

Foruden UTF-8 vejledningen har Rigsarkivet udarbejdet andre vejledninger, der har betydning for produktion og aflevering af afleveringspakker:

- Quickguide – til produktion og test af en afleveringspakke med ASTA
- Vejledning til bilag 9 om afleveringspakker i bekendtgørelse om arkiveringsversioner nr. 128
- Brugervejledning til ASTA
- Vejledning til produktion af afleveringspakke med data fra regneark eller csv-filer
- Vejledning til Skab archiveIndex
- Vejledning til Skab contextDocumentationIndex
- Vejledning om konvertering af dokumenter til TIFF
- Eksempelafleveringspakke med statistikdata FD.18005

Alt vejledningsmateriale kan tilgås fra Rigsarkivets hjemmeside www.sa.dk.

C. Lovgivning og retsforskrifter

Information om lovgivning m.v. findes på Rigsarkivets hjemmeside www.sa.dk.

D. Definitioner

Afleveringspakke med data fra statistikfiler består overordnet set af kontekstdokumenter, der skal afleveres i Rigsarkivets arkivformater, udtræk af data og metadata fra de statistikfiler, som skal afleveres, samt to indeksfiler i xml-format, der indeholder overordnet metadata om de afleverede data og kontekstdokumenterne.

1. Hvad er UTF-8 indkodning?

Ord og sætninger i tekst består af tegn, som er de bogstaver, som man kan se, fx a, b, c, å og @. Computeren bruger sit eget 'sprog', som repræsenterer alle tegn, tal og bogstaver i form af bytes som består af 8 bits og hver byte kan antage en af værdierne mellem 0-255.

Når man gemmer en almindelig tekstfil, så tildeles hver bogstav et tal (fx 'A' tildeles værdien 65, 'B' værdien 66 etc.) og disse tal gemmes herefter på computerens harddisk. Når tekstfilen skal læses igen, så sørger programmerne for at oversætte og vise disse talværdierne som tegn på skærmen (Fx værdien 65 bliver vist som 'A').

Den nok ældste og mest almindelige mapping mellem talværdier og karakterer kaldes ASCII og den fulde mapping tabel kan ses i figur 1.1.

ASCII value	Character	ASCII value	Character	ASCII value	Character
000	^@	043	+	086	V
001	^A	044	,	087	W
002	^B	045	-	088	X
003	^C	046	.	089	Y
004	^D	047	/	090	Z
005	^E	048	0	091	[
006	^F	049	1	092	\
007	^G	050	2	093	]
008	^H	051	3	094	^
009	^I	052	4	095	_
010	^J	053	5	096	`
011	^K	054	6	097	a
012	^L	055	7	098	b
013	^M	056	8	099	c
014	^N	057	9	100	d
015	^O	158	:	101	e
016	^P	059	;	102	f
017	^Q	060	<	103	g
018	^R	061	=	104	h
019	^S	062	>	105	i
020	^T	063	?	106	j
021	^U	064	@	107	k
022	^V	065	A	108	l
023	^W	066	B	109	m
024	^X	067	C	110	n
025	^Y	068	D	111	o
026	^Z	069	E	112	p
027	^[	070	F	113	q
028	^\	071	G	114	r
029	^]	072	H	115	s
030	^^	073	I	116	t
031	^-	074	J	117	u
032	[space]	075	K	118	v
033	!	076	L	119	w
034	"	077	M	120	x
035	#	078	N	121	y
036	$	079	O	122	z
037	%	080	P	123	{
038	&	081	Q	124	|
039	'	082	R	125	}
040	(	083	S	126	~
041	)	084	T	127	DEL
042	*	085	U		

Figur 1.1 ASCII tabel med oversættelse mellem tegn og decimale værdier (ASCII value)

I computerens tidligste barndom var understøttelse af nationale karakterer så og sige "en by i Rusland" og man måtte nøjes med de amerikanske karakterer. Dette ændrede sig da andre lande begyndte at udskifte visse (mindre betydende) karakterer i ASCII tabellen til fordel for Æ, Ø og Å. Disse udgaver af ASCII blev fx kaldt Code page 865 (Nordic languages). Problemet var (og er) at man ikke anede om en ASCII tekstfil er gemt på det ene eller det andet sprog når filen skulle vises, og man måtte ofte prøve sig frem.

Denne problematik blev gennem årene forsøgt løst ved at finde på nye tekstformater fx ANSI, EBCDIC, og Unicode. Den seneste og bedste løsning kaldes UTF-8 som kan benyttes til alle sprog fordi den benytter 1-4 bytes og derfor kan repræsentere et meget stort antal karakterer (herunder Æ, Ø og Å).

2. Hvad er konsekvensen, hvis data ikke er indkodet som UTF-8 tegnsæt?

I forbindelse med aflevering af statistikfiler i form af en afleveringspakke til arkivet er det den afleverende myndigheds ansvar at udtrække data og metadata fra statistikfilerne til afleveringspakken. Det er den, der producerer afleveringspakken, der skal sikre, at alle tegn er korrekt indkodet som UTF-8 før udtræk.

Hvis et datasæt oprindeligt stammer fra en af de nyere versioner af SAS, SPSS eller STATA, er tegnsættet i statistikfilen højst sandsynligt indkodet som UTF-8, fordi nyere versioner af statistikprogrammerne anvender denne som default indkodning. Hvis statistikprogrammets opsætning ikke er Unicode, kan du ændre denne opsætning i 'Preference'/'Options' i statistiskprogrammet og derefter gemme filen som Unicode før udtræk af data til afleveringspakken via programmet ASTA og dermed sikre dig, at alle tegn vises korrekt.

Hvis dit datasæt stammer fra et andet program eller oprindeligt har en anden indkodning fx ANSI og importeres til et statistikprogram, som anvender Unicode som default, kan dette forårsage at nogle tegn vises forkert fx et ord som 'stå' kan vises som 'st□', da transformationen kan påvirke tegnene. Er dette sket, er det vigtigt at rette tegnene, som er forkerte, så andre i fremtiden kan bruge, læse og forstå datasættet.

Rigsarkivets værktøj ASTA, der kan anvendes til test af afleveringspakken før aflevering til arkiv, tester ikke automatisk for tegnsættet i data- og metadatafilen. Rigsarkivet tester visuelt alle afleverede data- og metadatafiler efter aflevering. Hvis der er ugyldige tegn i afleveringen, som ikke er UTF-8 tegn, får arkiverskaber besked herom, og der kræves nogle justeringer i datafilen. Typisk vil nye dataudtræk og en genaflevering være nødvendig.

Derfor er det vigtigt, at du visuelt tjekker din datafil udtrukket til afleveringspakken for at sikre dig, at alle tegn vises korrekt og kan tydes. Læs mere herom i afsnit 5 og 6.

3. Hvordan aflæses og ændres statistikfilers indkodning i statistikprogrammer?

Sørg for at kontrollere, at statistikfilens indkodning er UTF-8, før du laver udtræk af data til afleveringspakken.

Hvert statistikprogram har sin egen syntaks til at undersøge tegnsættet i et datasæt og ændre dette til et andet tegnsæt. Nedenfor finder du procedurer og syntakser for SAS, Stata og SPSS. Ved brug af disse kan du sikre dig, at data er indkodet som UTF-8 tegnsæt.

A. SAS – syntakser og procedurer til tegnsæt

Undersøg SAS-filens indkodning/tegnsæt

For at identificere datasættes indkodning/tegnsæt i en SAS-datafil, skal du følge disse trin, som er anbefalet af SAS[1]:

- Kør følgende SAS-syntaks for at bestemme indkodningen/tegnsættet for et datasæt i SAS. Du skal kun erstatte libref.data_set_name med dit biblioteks navn og filnavn (for eksempel "mylib.mydata").

SAS-Syntaks

```
%let dsn=libref.data_set_name;
%let dsid=%sysfunc(open(&dsn,i));
%put &dsn ENCODING is: %sysfunc(attrc(&dsid,encoding));
```

Eksempel

```
%let dsn=dgi.customerdaga;
%let dsid=%sysfunc(open(&dsn,i));
%put &dsn ENCODING is: %sysfunc(attrc(&dsid,encoding));
```

En anden måde at finde SAS-filens indkodning/tegnsæt er at køre en ”proc contents”, som i output viser filens indkodning/tegnsæt. Sådan:

SAS-Syntaks

```
PROC CONTENTS <option-1 <...option-n>>;
run;
```

Eksempel

```
PROC CONTENTS  data=dgi.customerdata;
```

Skift SAS-filens indkodning/tegnsæt til UTF-8:

For at ændre indkodning/tegnsæt af en SAS-fil, der ikke tidligere har været defineret som UTF-8, kan nedenstående syntaks anvendes[2]. Bemærk at du skal erstatte følgende:

1) libref samt dets placering
2) Angiv den placering, du ønsker at gemme den nye UTF-8-fil i (Syntaksens anden linje)
3) Angiv det ønskede datasætnavn til den nye UTF-8-fil (Syntaksens fjerde linje)

SAS-Syntaks

```
libname inlib libref 'c:\xxxx';
libname outlib 'c:\yyy' outencoding='UTF-8';
proc copy noclone in=inlib out=outlib;
select dataset_name;
run;
```

[1] http://support.sas.com/kb/14/290.html

[2] http://support.sas.com/kb/15/597.html

Eksempel
libname inlib dgi 'c:\temp';
libname outlib 'c:\temp\out' outencoding='UTF-8';
proc copy noclone in=inlib out=outlib;
select customerdata;
run;

B. SPSS – syntakser og procedurer til tegnsæt

Indkodning/tegnsæt i forskellige SPSS-versioner:
Som beskrevet af IBM SPSS[3]:

- Frem til version 15 er alt indkodning/tegnsæt i SPSS baseret på code pages.
- Fra version 16 til 20 er Unicode (som UTF-8) også understøttet. UTF-8 er kaldt "**Unicode mode"** i SPSS 16. Bemærk at UTF-8 indkodning er både understøttet i datasæt og i syntaksfiler.
- Fra SPSS-version 21 og derefter spørger programmet, om "Unicode mode" skal anvendes, når det startes.

Undersøg SPSS-filens indkodning/tegnsæt
Følgende syntax kan køres i SPSS for at identificere om SPSS opsætning er i Unicode.

SPSS-Syntaks
SHOW UNICODE

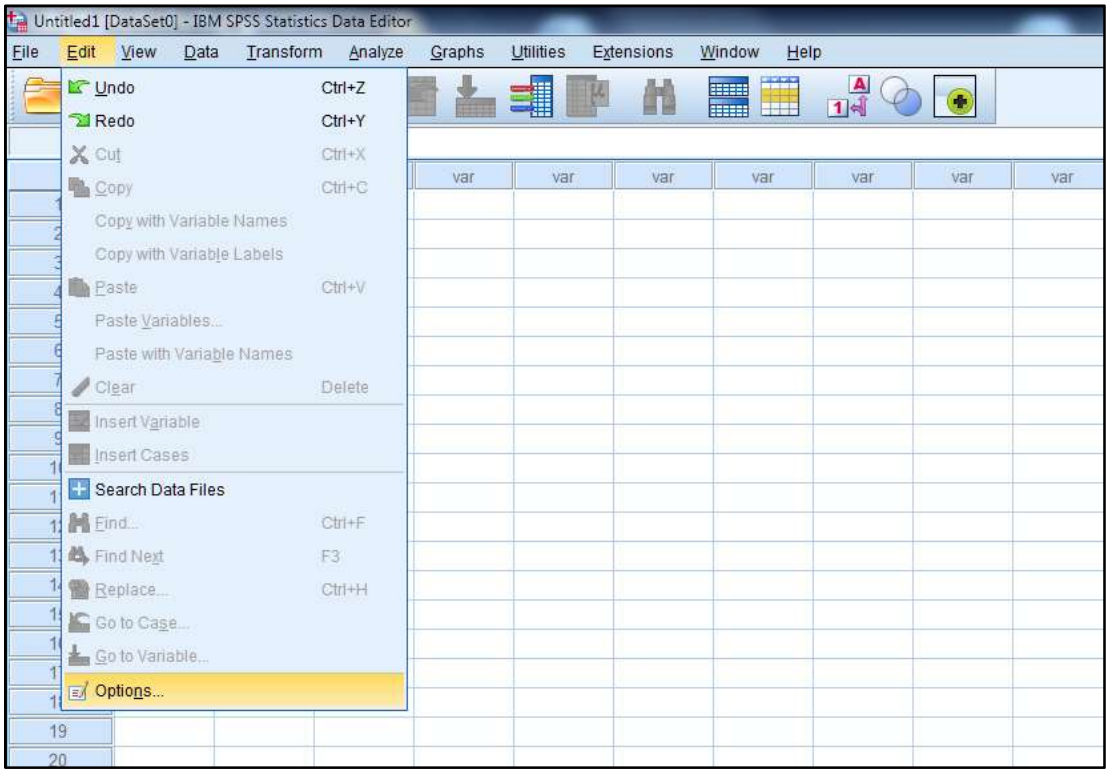
Undersøg og skift SPSS-filens indkodning/tegnsæt
For at identificere og skifte indkodning i SPSS, skal du åbne SPSS, og før du åbner datasættet, som du vil undersøge, skal du klikke på "Edit" og vælge 'Options' i menuen (se figur 1).

Et vindue åbnes nu med alle "Options". Vælg fanebladet "*Language"* (se figur 2). Marker "*Unicode (universal character set*)" for at vælge UTF-8 som SPSS default indkodning/tegnsæt for data og syntakser. Klik på '*OK'*.

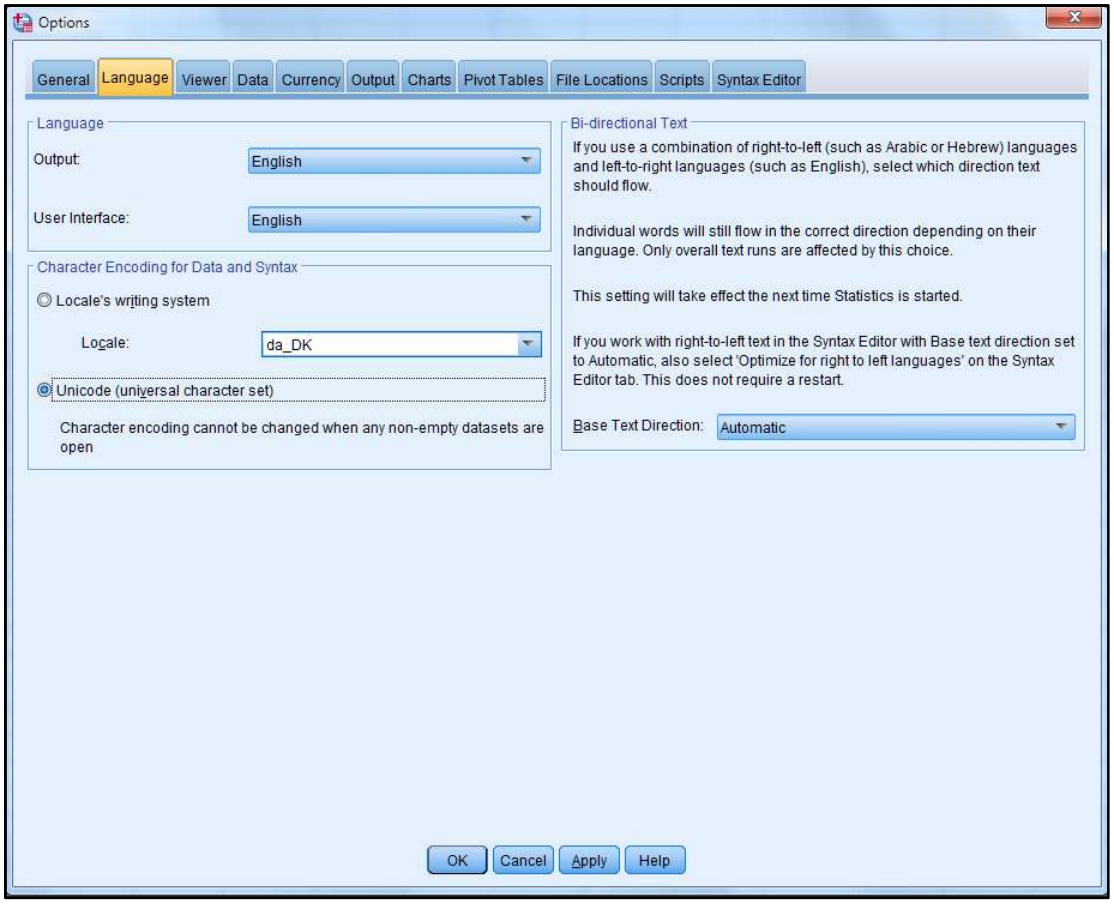
Du kan kontrollere, om ændringen er sket ved at kigge i bunden til højre af SPSS program-vinduet, som skal vise '*Unicode: ON'* (se figur 3).

Hvis der i feltet markeret med rød cirkel i figur 3 vises '*Unicode:OFF*', betyder det, at der i "language settings" (se figur 2) er markeret '*Locale's writing systems*' og ikke '*Unicode (universal character set)*'.

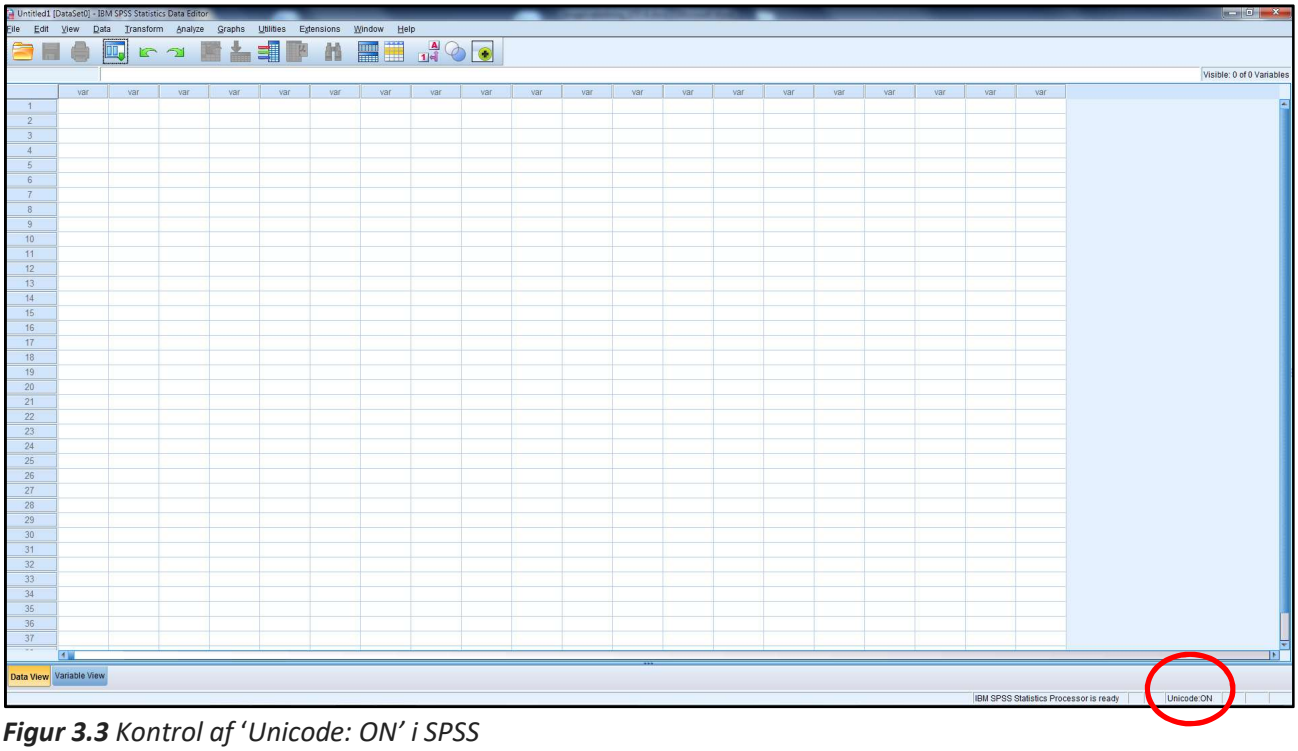
[3] https://www.spss-tutorials.com/spss-unicode-mode/



Figur 3.1 *Valg af Edit > Option i SPSS*



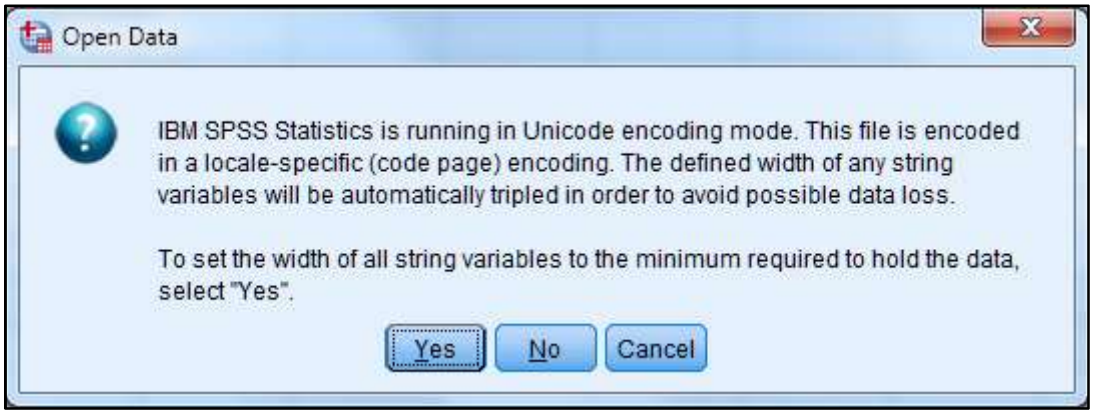
Figur 3.2 *Fanebladet "Language" under Options i SPSS*



Figur 3.3 *Kontrol af 'Unicode: ON' i SPSS*

Skift SPSS-filens indkodning til UTF-8:

Hvis du i SPSS åbner en fil, som ikke er indkodet som Unicode, efter du har aktiveret Unicode i SPSS, fremkommer et pop-up vindue med følgende besked:



Figur 3.4 *Pop-up vindue i SPSS*

Du skal vælge 'Yes' for at optimere antallet af bytes. Filen er nu gemt som UTF-8 format.

C. Stata – syntakser og procedurer til tegnsæt

Indkodning/tegnsæt i forskellige Stata-versioner:

Som beskrevet af Stata[4]:

- Stata 13 og tidligere versioner bruger ASCII som standard til indkodning/tegnsæt.
- I Stata 14 og efterfølgende versioner er UTF-8 default indkodning/tegnsæt til datasæt, do-filer, ado-filer og 'help' filer.

[4] https://www.stata.com/manuals/dunicodeencoding.pdf

Undersøg Stata-filens indkodning/tegnsæt

Til at analysere Stata-filens indkodning/tegnsæt i Stata, skal du køre følgende syntaks:

Stata-Syntaks
unicode analyze datasetname.dta

Eksempel
unicode analyze customerdata.dta

Skift Stata-filens indkodning/tegnsæt til UTF-8

Stata kan også oversætte filer fra 'extended ASCII' indkodning til Unicode (UTF-8). Først skal du definere, hvilken indkodning/tegnsæt du ønsker at oversætte filen til. Dette kan gøres ved at køre følgende syntaks:

Stata-Syntaks
unicode encoding set encodingnavn

Eksempel
unicode encoding set unicode

Dernæst kan du bruge følgende syntaks til at transformere Stata-filen til Unicode:

Stata-Syntaks
unicode translate myfile.dta

Eksempel
Unicode translate customerdata.dta

Hvis du kender source-filens (srcencoding) indkodning/tegnsæt og den indkodning, du ønsker at transformere den til (dstencoding), kan du anvende følgende syntaks:

Stata-Syntaks
unicode convertfile srcfilename destfilename , options

Eksempel
unicode convertfile " C:\Temp\customerdata.txt" "
C:\Temp\customerdata2.txt", srcencoding(ANSI1251)
dstencoding(UNICODE)

D. R og RStudio – syntakser og procedurer til tegnsæt

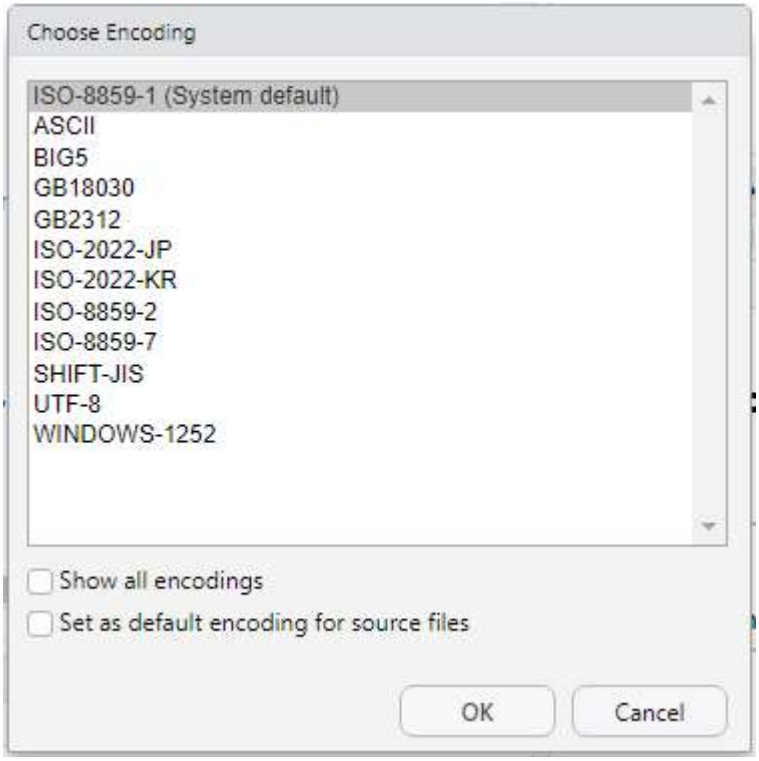
Indkodning i RStudio[5]

Fra og med RStudio version 0.93, understøtter RStudio, via 'platform native', indkodning af alle Unicode-tegn. Dvs. at programmet giver dig mulighed for at læse og skrive filer ved hjælp af en hvilken som helst tegnkodning, der er tilgængeligt på systemet. Du kan gøre dette ved at:

-vælge indkodning til læsning af filer ved at klikke på 'File'> 'Reopen with encoding', som genlæser den valgte fil fra disken med den nye kodning.

-gemme en åben fil med indkodningen ved at klikke på 'File' > 'Save with encoding'.

Kommandoerne *Genåbn* og *Gem med kodning* viser begge følgende dialog, hvor du skal vælge den ønskede kodning til at læse eller gemme en fil. Hvis det drejer sig om en aflevering til Rigsarkivet, skal man valg UTF-8 fra listen.



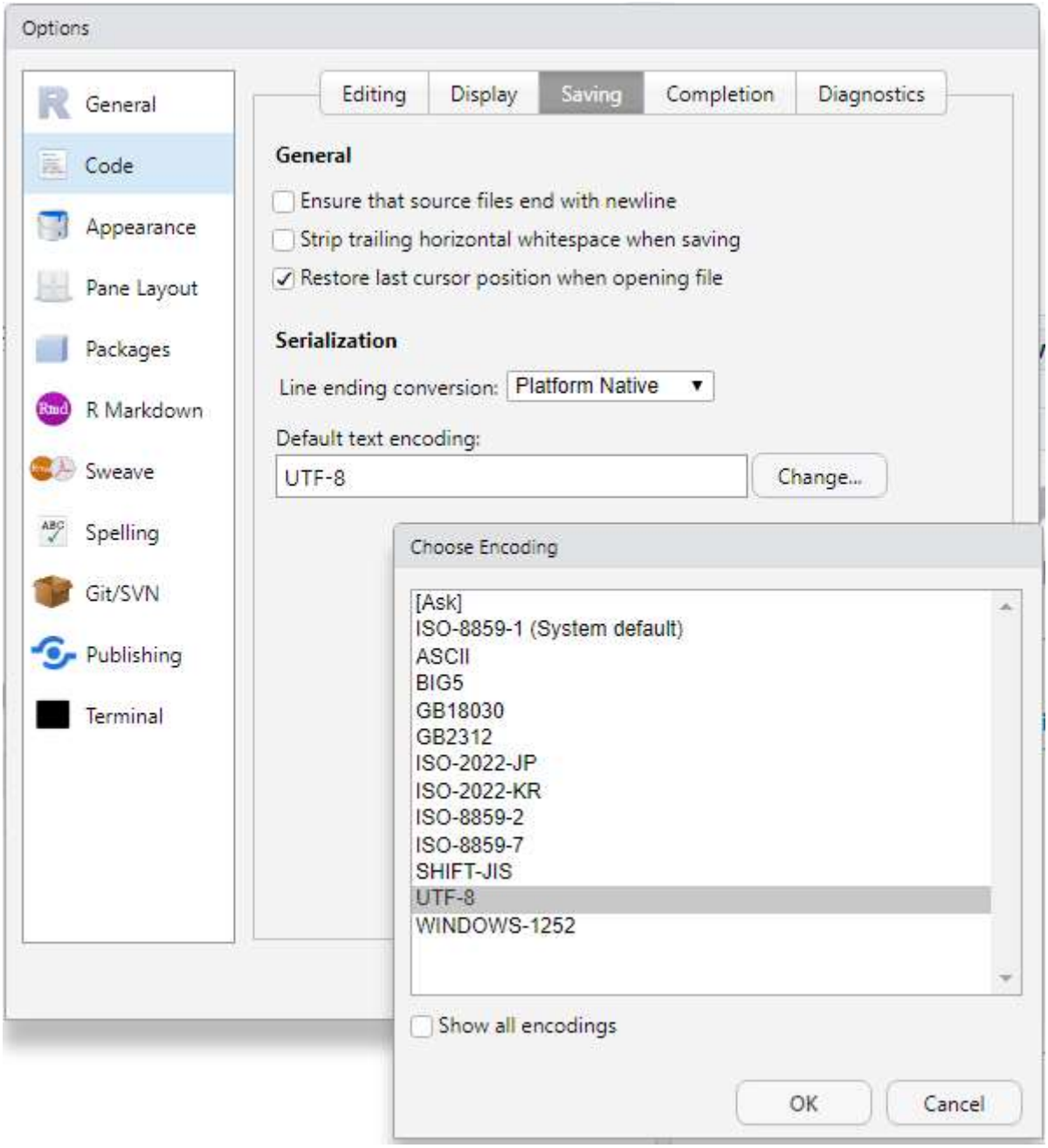
Ændring af indkodning i RStudio permanent

Hvis du vil ændre din standardkodning permanent i programmet, kan du gøre dette ved at:

1) klikke på 'Tools' > 'General Options';
2) vælge 'Code' i menu til venstre;
3) vælge fanen 'Saving' fra toppen (se nedenfor);
4) og klikke på 'Change'.

[5] https://support.rstudio.com/hc/en-us/articles/200532197-Character-Encoding

Når du klikker på 'Change', åbnes et popup-vindue, der giver dig mulighed for enten at vælge en anden standardkodning, eller at vælge "[ASK]" for at vælge en indkodning. Sådan:



Indkodninger i R

At prøve at ændre indkodningen til UTF-8 er en kompliceret opgave i R, fordi det varierer afhængigt af operativsystemet på din computer. For yderligere information om indkodninger i R, kan du læse mere om emnet og kompleksitet i artiklen "Escaping from character encoding hell in R on Windows" på dette websted https://dss.iq.harvard.edu/blog/escaping-character-encoding-hell-r-windows

4. Hvordan kontrolleres tegnsæt i en tekstfil?

Når du har udtrukket data og metadata fra statistikfilen til .csv- og .txt-filer, der overholder afleveringspakkens format for datafiler og metadatafiler, skal du også kontrollere, at alle tegn kan læses korrekt og er UTF-8 tegn.

Den udtrukne datafil (fx table1.csv) kan indeholde forkerte tegn, hvis den statistikfil, udtrækket er lavet fra, ikke var i UTF-8 (Unicode) før udtræk, eller hvis den oprindeligt indeholdt ikke gyldige UTF-8 tegn.

Du kan inspicere .csv- og .txt-filer i afleveringspakken for forkerte tegn på følgende måde:

- Find placeringen af din afleveringspakke. Mappen kaldes fx FD.12345
- Find datafilen, du vil kontrollere for ikke gyldige UTF-8 tegn, fx table1.csv ved at klikke ned i mappestrukturen: FD.12345 > Data > table1 > table1.csv
- Højreklik på 12345.csv og vælg 'åben med' fra popup-listen. Vælg at åbne filen med en teksteditor, fx *Notesblok* eller *Notepad++.*

  > **OBS**: Du skal ikke dobbeltklikke på filen for at åbne den, da dette automatisk kan åbne den op i Excel. Excel gætter ofte på tegnsæt og formaterne af dine data og kan derfor indlæse dine data forkert).

- Kontrollér indholdet af datafilen ved at kigge efter tegn, der ser mærkelige ud. Søg efter tegn som æ, ø og å, da disse ofte vises forkert, hvis tegnsættet ikke er UTF-8.
- Hvis du finder forkerte tegn, skal disse rettes i din originale datafil. Efter korrektionen skal et nyt udtræk laves (fx med programmet ASTA), og den udtrukne datafil skal igen visuelt testes for læsbarhed og ikke gyldige UTF-8 tegn.

OBS: En tekstfil med æ, ø og å indkodet som ANSI tegnsæt vil vise tegnene æ, ø og å korrekt i en teksteditor som Notepad. Ovenstående metode kan altså ikke med sikkerhed afdække om tekstfilen er indkodet som UTF-8 med gyldige UTF-8 tegn, men den kan afdække hvis tekstfilen indeholder tegn med en indkodning som ikke er i overensstemmelse med den indkodning teksteditoren anvender til visning af tegnene i tekstfilen.

Hvis du vil identificere tekstfilens indkodning, kan du aflæse hex-værdierne for tegnene i en binær fileditor.

5. Hvordan aflæses UTF-8 hex-værdier i en tekstfil?

Hvis du ønsker at vide præcis, om et tegn er et gyldigt UTF-8 tegn, kan du undersøge det binære indhold af et tegn i tekstfilen.

Til det formål skal du anvende en binær fileditor, fx HxD-fil. Denne Hex-editor viser både den binære, hexadecimale og decimale værdi af et tegn. Gyldige hexadecimale UTF-8 værdier for æ, ø, å, Æ, Ø og Å fremgår af figur 5.1.

Da HxD programmet ikke kan vise tegnrepræsentationen af en UTF-8 indkodning, kan du desværre ikke se selve tegnet vist korrekt (under 'Decoded text'). Derimod kan du få vist tegnene som de ser ud i ANSI, ASCII, Macintosh eller EBCDIC. Men du kan stadig se filens korrekte hexadecimale repræsentation af tegnet (se figur 5.3). Når du markerer et tegn i teksten til højre (tekst vist i ANSI format), markeres tegnets tilsvarende hexadecimale repræsentation af det binære tal, som repræsenterer tegnet. Marker specialtegn som æ, ø og å for at sikre dig, at disse specialtegn er i korrekt UTF-8 indkodning.

æ = **C3 E6** (vist i HxD som Ã¦)

ø = **C3 B8** (vist i HxD som Ã¸)

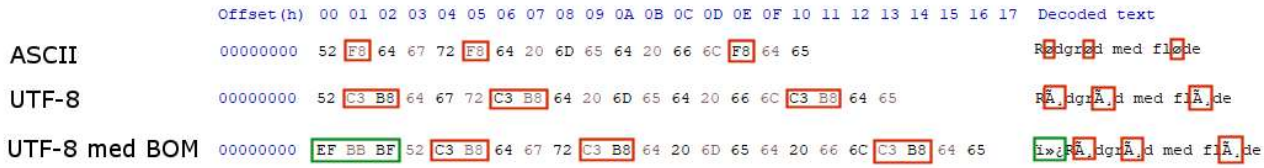
å = **C3 E5** (vist i HxD som Ã¥)

Æ = **C3 86** (vist i HxD som Ã†)

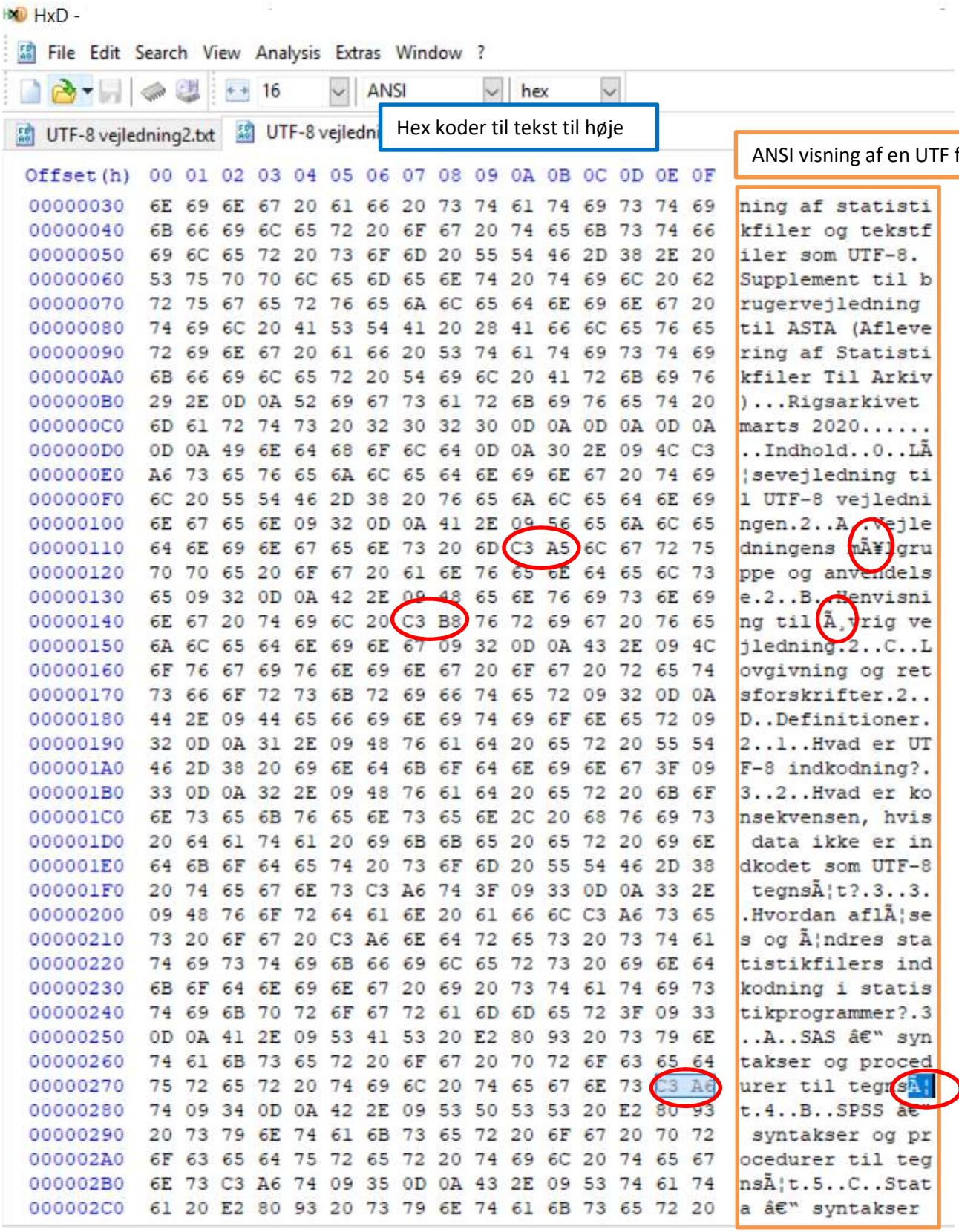
Ø = **C3 98** (vist i HxD som Ã˜)

Å = **C3 85** (vist i HxD som Ã…)

Figur 5.1 *Korrekte hexadecimale repræsentationer af æ, ø, å, Æ, Ø og Å i en tekstfil indkodet som UTF-8*



Figur 5.2 *Sammenligning af hexadecimale repræsentationer i ASCII og UTF-8*



Figur 5.3 *Visning af tegn og binære værdier af tegn i en hexeditor fra en UTF-8 indkodet fil*

6. Teknisk uddybning af UTF-8 indkodning, BOM og repræsentationer af tegn

E. Mere om UTF-8 indkodning

UTF-8 står for "Unicode Transformation Format". UTF-8 er en af de tre standard indkodninger af tegnsæt som bruges til at repræsentere Unicode som computertekst (de andre er UTF-16 og UTF-32). UTF-8 bruger en algoritme til at dekode data imellem en binær form (fx 01100001), der bruges af computere, og tegn (fx a) som kan læses af mennesker. '8' i UTF-8 betyder, at indkodningen bruger 8-bit blokke (en byte) til at repræsentere tegn. UTF-8 kan benyttes til alle sprog fordi den benytter mellem 1-4 bytes til at repræsentere et tegn og derfor kan repræsentere et meget stort antal karakterer (herunder Æ, Ø og Å). ASCII og ANSI benytter kun én byte pr. tegn.

Da UTF-8 er er en effektiv måde at lagre Unicode-tekst på og understøtter mange forskellige sprog har det medført at det er den mest anvendte Unicode-indkodning i dag.

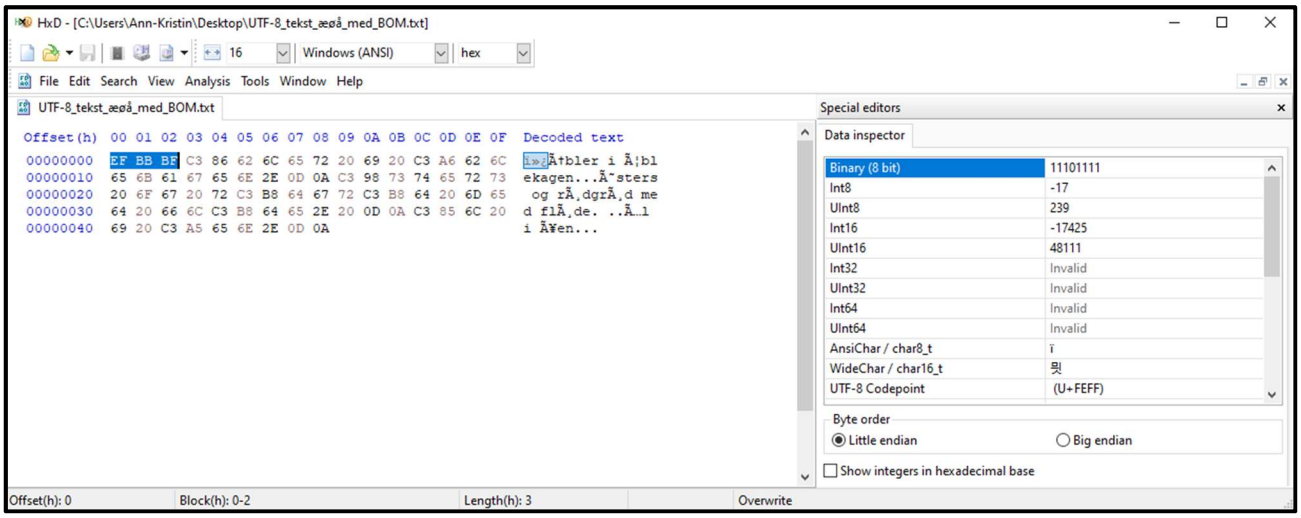
Dengang UTF-8 blev defineret ønskede man at UTF-8 skulle være bagud kompatibel med ASCII (se afsnit 1). De vigtigste 127 amerikanske bogstaver og tal i UTF-8 tabellen er derfor identiske med ASCII tabellens karakterer og fylder kun én byte. Andre landes nationale karakterer benytter 2 bytes (bogstavet 'Æ' har fx fået tildelt værdierne 195 og 166). Bogstaverne a-z repræsenteres således både i ASCII, ANSI og i UTF-8 med en enkelt byte med samme værdi. Det vil sige at en ANSI fil *uden* æ, ø og å også er en valid UTF-8 fil.

Når man gemmer en tekstfil kan man i nogle programmer selv vælge den ønskede indkodning for tegnsættet i filen, fx UTF-8. Andre gange gemmes tekstfilen blot med tekstprogrammets default indkodning/tegnsæt. Det er også muligt i nogle teksteditorer at konvertere mellem forskellige tegnsæt, fx at gemme en ANSI indkodet fil med UTF-8 indkodning. UTF-8 indkodning og dekoding af en tekstfil er således ikke altid noget man selv udfører, men noget som de programmer man benytter til at gemme (indkodning) og vise (dekoding) filen skal understøtte. Når programmer oversætter de binære repræsentationer af tegnene (fx 01100001) til læselige tegn (fx a) vil programmerne typisk forsøge at gætte på den rigtige indkodning af tekstfilen ved at lede efter UTF-8 tegn og derved forsøge at vise indholdet korrekt, hvilket typisk går godt, men som også indimellem kan fejle.

F. BOM (Byte Order Mark)

En UTF-8 fil kan tilføres 3 helt bestemte byteværdier i starten af filen med de hexadecimale værdier **EF BB BF**, se figur 1. Dette er et såkaldt Byte Order Mark også kaldet BOM. Når en tekstfil har denne BOM kan man være nogenlunde sikker på, at der er tale om en UTF-8 indkodning, men desværre er BOM mærket ikke påkrævet. Ofte er det heller ikke muligt at bestemme om programmet skal tilføje en BOM eller ej når filen gemmes.

Da det er muligt at kopiere en BOM ind i en tekstfil i en binær editor er tilstedeværelsen af en BOM ikke altid ensbetydende med at filen er indkodet som UTF-8, med mindre der også er gyldige UTF-8 hexadecimale værdier i filen (se afsnit 5).



Figur 6.1 *UTF-8 fil med BOM (hexadecimale værdier: EF BB BF)*

G. Forskellige repræsentationer af tegn

De tegn som en sætning består af fx a, b, c, å og @ eksisterer i computeren i binær form, dvs. i form af bytes. En byte består i UTF-8 af 8 bits og en bit kan enten have værdien 0 eller 1. Den binære værdi 01100001 repræsenterer fx bogstavet 'a' i både et ASCII, ANSI og UTF-8 tegnsæt. Men et tegn kan også repræsenteres af mere end en byte, fx bogstavet 'æ' som repræsenteres af de 2 bytes 11000011 10100110 i UTF-8 tegnsættet.

Der findes mange forskellige indkodninger som kan oversætte computerens bytes til tegn. Eksempler på forskellige indkodninger af tegnsæt er ASCII, ANSI, EBCDIC, Unicode og UTF-8. Afhængigt at den valgte indkodning i en teksteditor tolker (dekoder) computeren de bytes der findes i tekstfilen forskelligt og viser dermed også tegnene forskelligt. Hvis teksteditorens valg af indkodning for filen (som skal anvendes til oversættelse/dekodning af de binære værdier) ikke stemmer overens med tekstfilen indkodning, bliver visualisering af teksten ødelagt, dvs. man ser ikke de korrekte tegn, fx å vises som □ eller [eller lignende.

Computeren gemmer et ord som 'hello' med 5 bytes (40bits): 01001000 01100101 01101100 01101100 01101111. Fordi disse binære numre ofte er lange og besværlige at vise i en tabel eller et program, bliver de ofte vist i andre former, som fx decimal, hexadecimal eller codepoints. I tabel 1 kan du se forskelige repræsentationer af bogstavet a i en tekstfil som er indkodet som UTF-8**.**

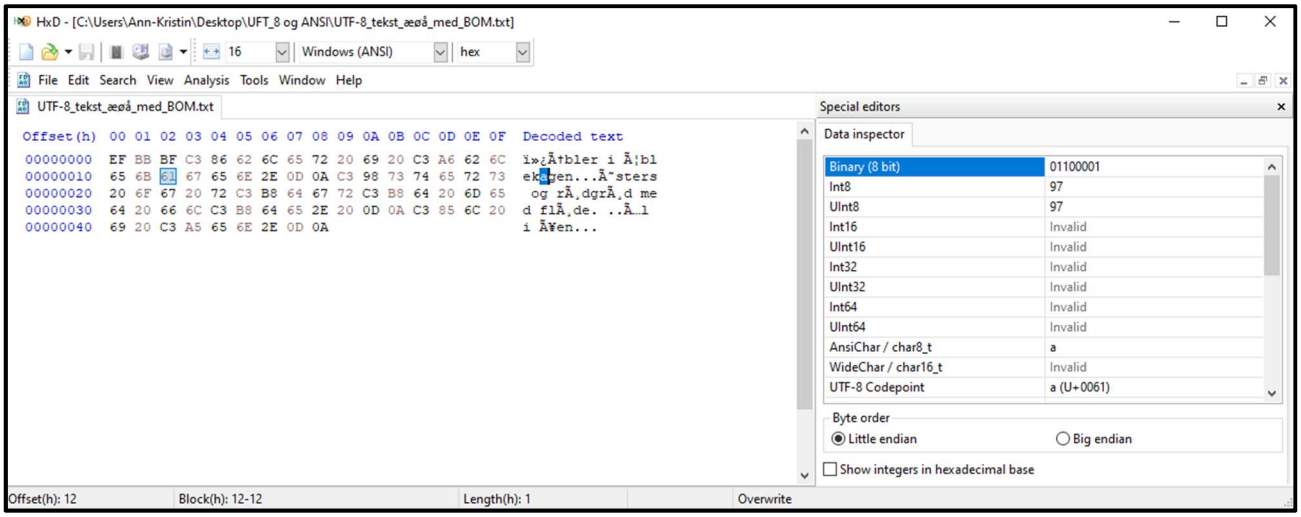
Det binære talsystem er et 2-talsystem bestående af de to tal 0 og 1. En binær værdi kan omregnes til en decimal værdi (i 10-talsystemet). Den decimale repræsentation af den binære værdi 01100001 for tegnet a udregnes på følgende måde 0x128 + 1x64 + 1x32 + 0x16 + 0x8 + 0x4 + 0x2 + 1x1 = 97.

Tabel 1. Forskellige UTF-8 repræsentationer af tegn

Tegn/ Bogstav	**Binær (UTF-8)** 128 64 32 16 8 4 2 1	**Decimal (UTF-8)**	**Hexadecimal (UTF-8)**	**UTF8 Codepoints**
A	01100001	97	61	U+0061
B	01100010	98	62	U+0062

Æ	11000011 10100110	195 166	C3 A6	U+00E6
Ø	11000011 10111000	195 184	C3 B8	U+00F8
Å	11000011 10100101	195 165	C3 A5	U+00E5
Æ	11000011 10000110	195 134	C3 86	U+00C6
Ø	11000011 10011000	195 152	C3 98	U+00D8
Å	11000011 10000101	195 133	C3 85	U+00C5
BOM			EF BB BF	U+FEFF

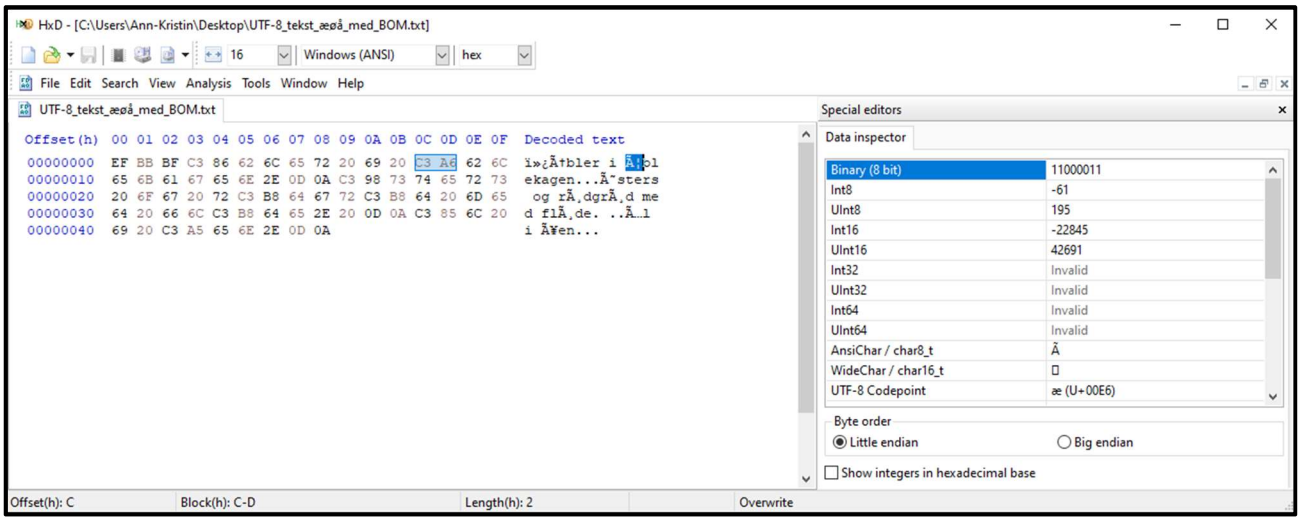
Flere af disse repræsentationer af tegnet kan ses binære teksteditorer som fx HxD, beskrevet nærmere i forrige afsnit og illustreret i nedenstående figurer.



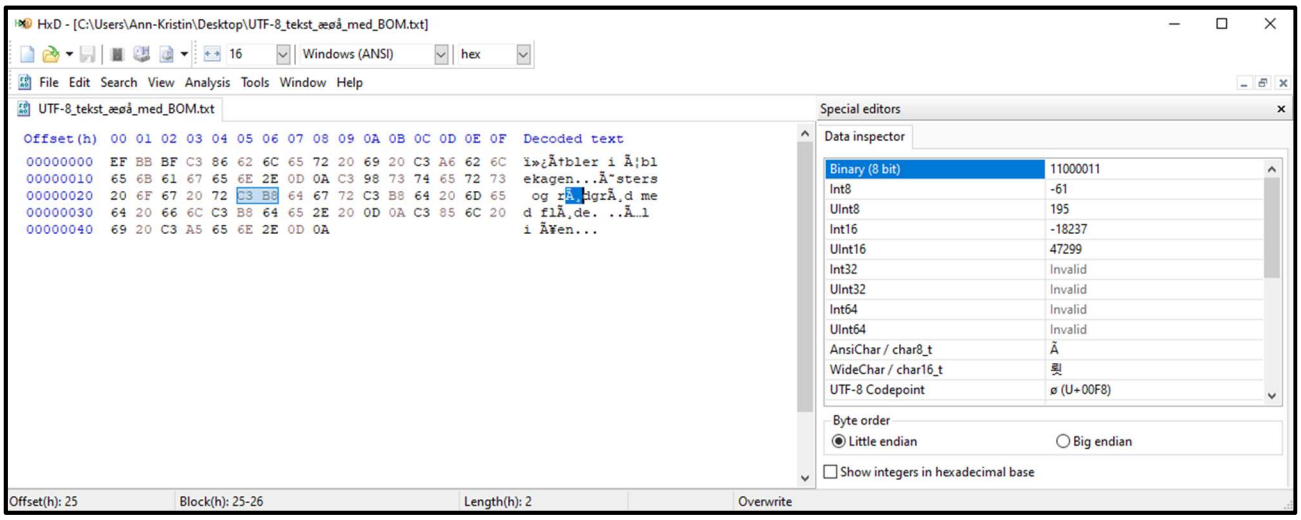
Figur 6.2 *UTF-8 fil med lille a (hexadecimale værdi: 61)*

BEMÆRK at selve tegnet i UTF-8 tekstfilerne i figurerne herunder ikke vises som æ, ø og å, da HxD editoren ikke kan decode/oversætte teksten til UTF-8 visning af tegnene. I stedet vises hvordan gyldige indkodede UTF-8 hexadecimale værdier oversættes/decodes til ANSI tegn (decoded text).

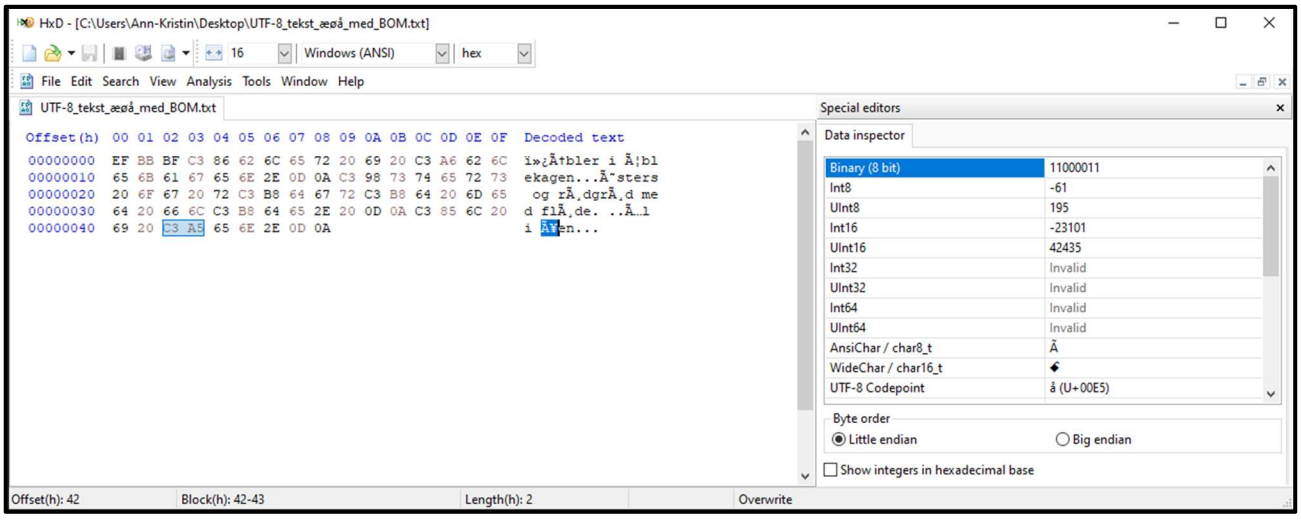
BEMÆRK at den binære værdi (Binary) kun vises for den første byte i tegn bestående af to bytes. De samme gælder den decimale værdi (UInt8). For at aflæse de binære og decimale værdier for hver enkelt byte skal hver byte markeres for sig.



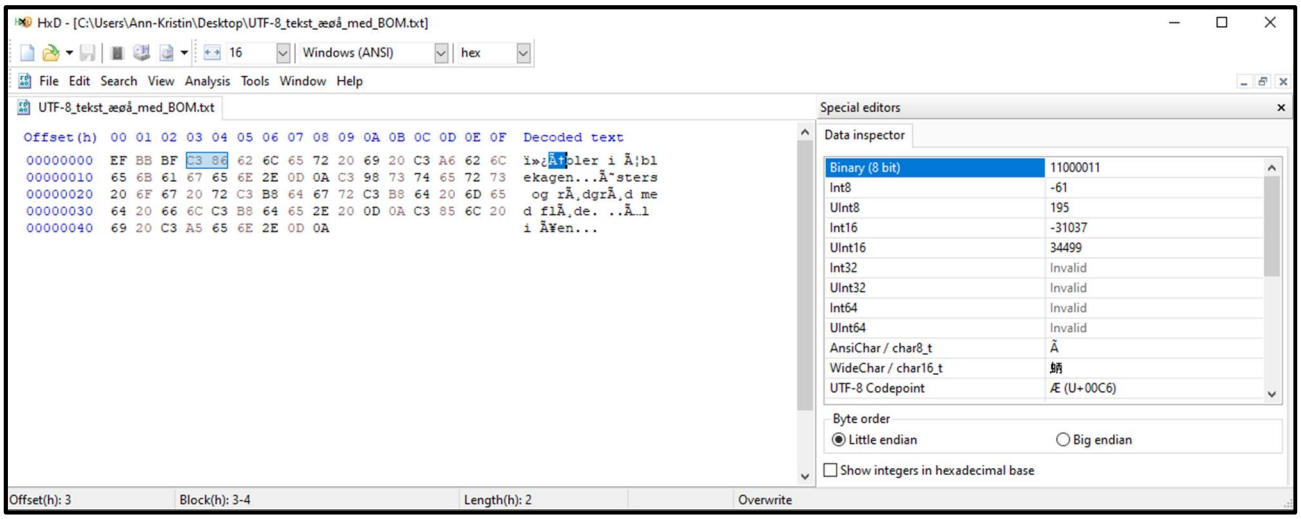
Figur 6.3 *UTF-8 fil med lille æ (hexadecimale værdier: C3 A6)*



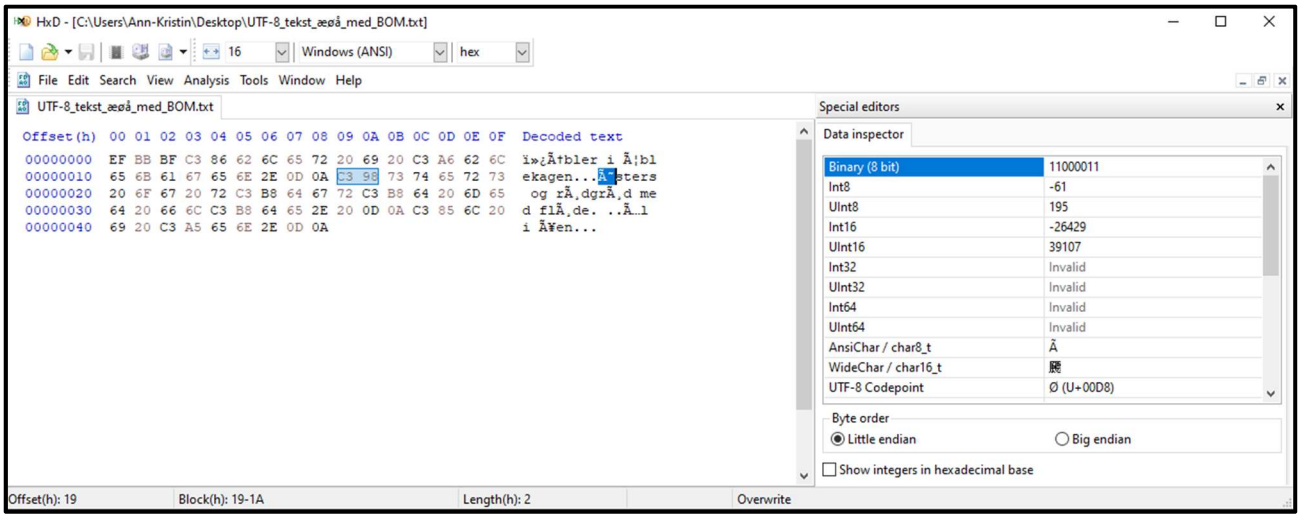
Figur 6.4 *UTF-8 fil med lille ø (hexadecimale værdier: C3 B8)*



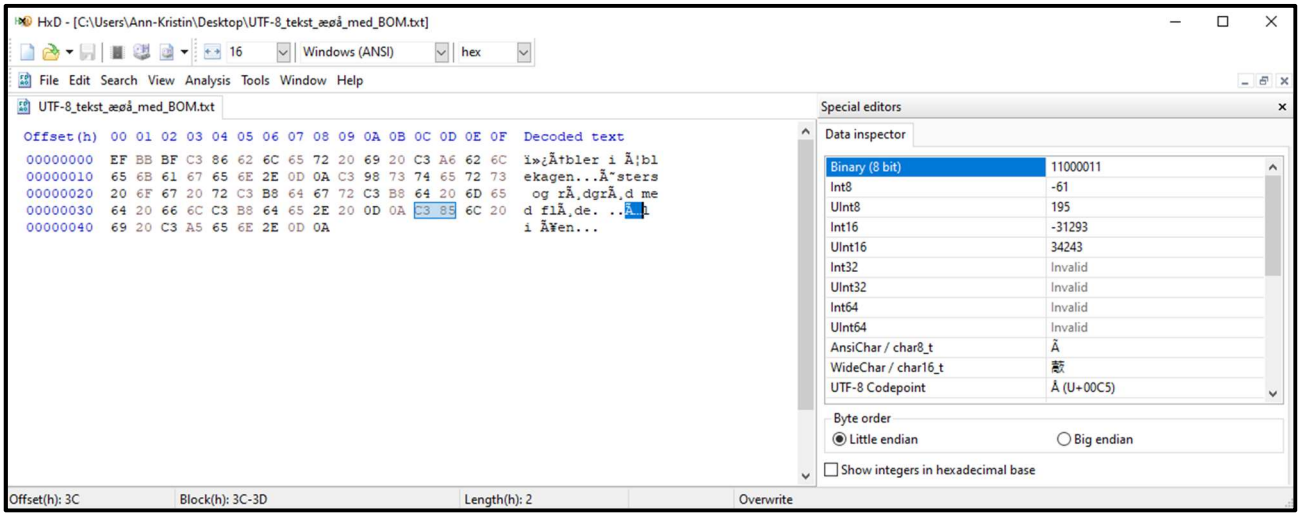
Figur 6.5 *UTF-8 fil med lille å (hexadecimale værdier: C3 A5)*



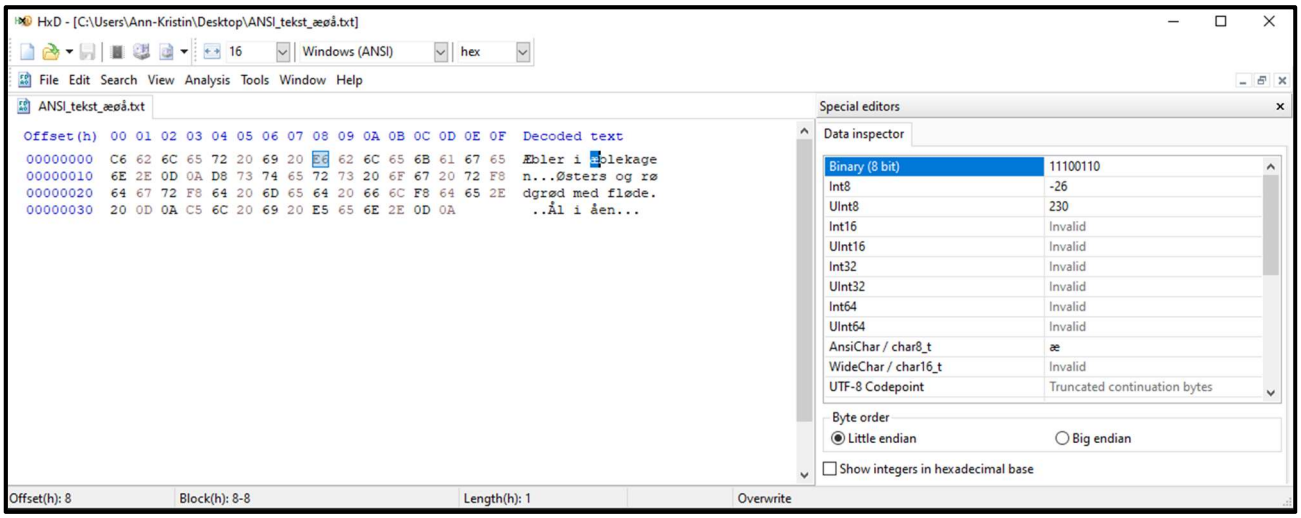
Figur 6.6 *UTF-8 fil med Stort Æ (hexadecimale værdier: C3 86)*



Figur 6.7 *UTF-8 fil med Stort Ø (hexadecimale værdier: C3 98)*



Figur 6.8 *UTF-8 fil med Stort Å (hexadecimale værdier: C3 85)*



***Figur 6.9** ANSI fil med lille æ (hexadecimale værdier: E6)*

BEMÆRK at figur 6.9 viser en fil indkodet med ANSI tegnsæt. Den hexadecimale værdi er E6 som er en reference til UTF-8 codepoint U+00E6 for lille æ i UTF-8. Den hexadecimale værdi E6 er *ikke* et gyldigt UTF-8 hexadecimal værdi for lille æ.

BEMÆRK at æ, ø og å vises korrekt i figur 6.9, fordi tekstfilen er indkodet som ANSI og teksteditoren oversætter/dekoder disse værdier til ANSI tegn (Decoded text).

7. UTF-8 support i Rigsarkivet

Hvis du oplever problemer med at identificere tegnsæt i filer og ændre tegnsæt til UTF-8, kan du kontakte datamanageren for forskningsdata i Rigsarkivet på følgende e-mail: mailbox@sa.dk .

Bilag 1 UTF-8 tabel med oversættelse mellem tegn og hex-værdi

Unicode code point	character	UTF-8 (hex.)	name
U+0020		20	SPACE
U+0021	!	21	EXCLAMATION MARK
U+0022	"	22	QUOTATION MARK
U+0023	#	23	NUMBER SIGN
U+0024	$	24	DOLLAR SIGN
U+0025	%	25	PERCENT SIGN
U+0026	&	26	AMPERSAND
U+0027	'	27	APOSTROPHE
U+0028	(	28	LEFT PARENTHESIS
U+0029	)	29	RIGHT PARENTHESIS
U+002A	*	2a	ASTERISK
U+002B	+	2b	PLUS SIGN
U+002C	,	2c	COMMA
U+002D	-	2d	HYPHEN-MINUS
U+002E	.	2e	FULL STOP
U+002F	/	2f	SOLIDUS
U+0030	0	30	DIGIT ZERO
U+0031	1	31	DIGIT ONE
U+0032	2	32	DIGIT TWO
U+0033	3	33	DIGIT THREE
U+0034	4	34	DIGIT FOUR
U+0035	5	35	DIGIT FIVE
U+0036	6	36	DIGIT SIX
U+0037	7	37	DIGIT SEVEN
U+0038	8	38	DIGIT EIGHT
U+0039	9	39	DIGIT NINE
U+003A	:	3a	COLON
U+003B	;	3b	SEMICOLON
U+003C	<	3c	LESS-THAN SIGN
U+003D	=	3d	EQUALS SIGN
U+003E	>	3e	GREATER-THAN SIGN
U+003F	?	3f	QUESTION MARK
U+0040	@	40	COMMERCIAL AT
U+0041	A	41	LATIN CAPITAL LETTER A
U+0042	B	42	LATIN CAPITAL LETTER B
U+0043	C	43	LATIN CAPITAL LETTER C
U+0044	D	44	LATIN CAPITAL LETTER D
U+0045	E	45	LATIN CAPITAL LETTER E

U+0046	F	46	LATIN CAPITAL LETTER F
U+0047	G	47	LATIN CAPITAL LETTER G
U+0048	H	48	LATIN CAPITAL LETTER H
U+0049	I	49	LATIN CAPITAL LETTER I
U+004A	J	4a	LATIN CAPITAL LETTER J
U+004B	K	4b	LATIN CAPITAL LETTER K
U+004C	L	4c	LATIN CAPITAL LETTER L
U+004D	M	4d	LATIN CAPITAL LETTER M
U+004E	N	4e	LATIN CAPITAL LETTER N
U+004F	O	4f	LATIN CAPITAL LETTER O
U+0050	P	50	LATIN CAPITAL LETTER P
U+0051	Q	51	LATIN CAPITAL LETTER Q
U+0052	R	52	LATIN CAPITAL LETTER R
U+0053	S	53	LATIN CAPITAL LETTER S
U+0054	T	54	LATIN CAPITAL LETTER T
U+0055	U	55	LATIN CAPITAL LETTER U
U+0056	V	56	LATIN CAPITAL LETTER V
U+0057	W	57	LATIN CAPITAL LETTER W
U+0058	X	58	LATIN CAPITAL LETTER X
U+0059	Y	59	LATIN CAPITAL LETTER Y
U+005A	Z	5a	LATIN CAPITAL LETTER Z
U+005B	[	5b	LEFT SQUARE BRACKET
U+005C	\	5c	REVERSE SOLIDUS
U+005D	]	5d	RIGHT SQUARE BRACKET
U+005E	^	5e	CIRCUMFLEX ACCENT
U+005F	_	5f	LOW LINE
U+0060	`	60	GRAVE ACCENT
U+0061	a	61	LATIN SMALL LETTER A
U+0062	b	62	LATIN SMALL LETTER B
U+0063	c	63	LATIN SMALL LETTER C
U+0064	d	64	LATIN SMALL LETTER D
U+0065	e	65	LATIN SMALL LETTER E
U+0066	f	66	LATIN SMALL LETTER F
U+0067	g	67	LATIN SMALL LETTER G
U+0068	h	68	LATIN SMALL LETTER H
U+0069	i	69	LATIN SMALL LETTER I
U+006A	j	6a	LATIN SMALL LETTER J
U+006B	k	6b	LATIN SMALL LETTER K
U+006C	l	6c	LATIN SMALL LETTER L
U+006D	m	6d	LATIN SMALL LETTER M
U+006E	n	6e	LATIN SMALL LETTER N
U+006F	o	6f	LATIN SMALL LETTER O
U+0070	p	70	LATIN SMALL LETTER P

U+0071	q	71	LATIN SMALL LETTER Q
U+0072	r	72	LATIN SMALL LETTER R
U+0073	s	73	LATIN SMALL LETTER S
U+0074	t	74	LATIN SMALL LETTER T
U+0075	u	75	LATIN SMALL LETTER U
U+0076	v	76	LATIN SMALL LETTER V
U+0077	w	77	LATIN SMALL LETTER W
U+0078	x	78	LATIN SMALL LETTER X
U+0079	y	79	LATIN SMALL LETTER Y
U+007A	z	7a	LATIN SMALL LETTER Z
U+007B	{	7b	LEFT CURLY BRACKET
U+007C	|	7c	VERTICAL LINE
U+007D	}	7d	RIGHT CURLY BRACKET
U+007E	~	7e	TILDE
U+00A0		c2 a0	NO-BREAK SPACE
U+00A1	¡	c2 a1	INVERTED EXCLAMATION MARK
U+00A2	¢	c2 a2	CENT SIGN
U+00A3	£	c2 a3	POUND SIGN
U+00A4	¤	c2 a4	CURRENCY SIGN
U+00A5	¥	c2 a5	YEN SIGN
U+00A6	¦	c2 a6	BROKEN BAR
U+00A7	§	c2 a7	SECTION SIGN
U+00A8	¨	c2 a8	DIAERESIS
U+00A9	©	c2 a9	COPYRIGHT SIGN
U+00AA	ª	c2 aa	FEMININE ORDINAL INDICATOR
U+00AB	«	c2 ab	LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
U+00AC	¬	c2 ac	NOT SIGN
U+00AD		c2 ad	SOFT HYPHEN
U+00AE	®	c2 ae	REGISTERED SIGN
U+00AF	¯	c2 af	MACRON
U+00B0	°	c2 b0	DEGREE SIGN
U+00B1	±	c2 b1	PLUS-MINUS SIGN
U+00B2	²	c2 b2	SUPERSCRIPT TWO
U+00B3	³	c2 b3	SUPERSCRIPT THREE
U+00B4	´	c2 b4	ACUTE ACCENT
U+00B5	µ	c2 b5	MICRO SIGN
U+00B6	¶	c2 b6	PILCROW SIGN
U+00B7	·	c2 b7	MIDDLE DOT
U+00B8	¸	c2 b8	CEDILLA
U+00B9	¹	c2 b9	SUPERSCRIPT ONE
U+00BA	º	c2 ba	MASCULINE ORDINAL INDICATOR

U+00BB	»	c2 bb	RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK
U+00BC	¼	c2 bc	VULGAR FRACTION ONE QUARTER
U+00BD	½	c2 bd	VULGAR FRACTION ONE HALF
U+00BE	¾	c2 be	VULGAR FRACTION THREE QUARTERS
U+00BF	¿	c2 bf	INVERTED QUESTION MARK
U+00C0	À	c3 80	LATIN CAPITAL LETTER A WITH GRAVE
U+00C1	Á	c3 81	LATIN CAPITAL LETTER A WITH ACUTE
U+00C2	Â	c3 82	LATIN CAPITAL LETTER A WITH CIRCUMFLEX
U+00C3	Ã	c3 83	LATIN CAPITAL LETTER A WITH TILDE
U+00C4	Ä	c3 84	LATIN CAPITAL LETTER A WITH DIAERESIS
U+00C5	Å	c3 85	LATIN CAPITAL LETTER A WITH RING ABOVE
U+00C6	Æ	c3 86	LATIN CAPITAL LETTER AE
U+00C7	Ç	c3 87	LATIN CAPITAL LETTER C WITH CEDILLA
U+00C8	È	c3 88	LATIN CAPITAL LETTER E WITH GRAVE
U+00C9	É	c3 89	LATIN CAPITAL LETTER E WITH ACUTE
U+00CA	Ê	c3 8a	LATIN CAPITAL LETTER E WITH CIRCUMFLEX
U+00CB	Ë	c3 8b	LATIN CAPITAL LETTER E WITH DIAERESIS
U+00CC	Ì	c3 8c	LATIN CAPITAL LETTER I WITH GRAVE
U+00CD	Í	c3 8d	LATIN CAPITAL LETTER I WITH ACUTE
U+00CE	Î	c3 8e	LATIN CAPITAL LETTER I WITH CIRCUMFLEX
U+00CF	Ï	c3 8f	LATIN CAPITAL LETTER I WITH DIAERESIS
U+00D0	Ð	c3 90	LATIN CAPITAL LETTER ETH
U+00D1	Ñ	c3 91	LATIN CAPITAL LETTER N WITH TILDE
U+00D2	Ò	c3 92	LATIN CAPITAL LETTER O WITH GRAVE
U+00D3	Ó	c3 93	LATIN CAPITAL LETTER O WITH ACUTE
U+00D4	Ô	c3 94	LATIN CAPITAL LETTER O WITH CIRCUMFLEX
U+00D5	Õ	c3 95	LATIN CAPITAL LETTER O WITH TILDE
U+00D6	Ö	c3 96	LATIN CAPITAL LETTER O WITH DIAERESIS
U+00D7	×	c3 97	MULTIPLICATION SIGN
U+00D8	Ø	c3 98	LATIN CAPITAL LETTER O WITH STROKE
U+00D9	Ù	c3 99	LATIN CAPITAL LETTER U WITH GRAVE
U+00DA	Ú	c3 9a	LATIN CAPITAL LETTER U WITH ACUTE
U+00DB	Û	c3 9b	LATIN CAPITAL LETTER U WITH CIRCUMFLEX
U+00DC	Ü	c3 9c	LATIN CAPITAL LETTER U WITH DIAERESIS
U+00DD	Ý	c3 9d	LATIN CAPITAL LETTER Y WITH ACUTE
U+00DE	Þ	c3 9e	LATIN CAPITAL LETTER THORN
U+00DF	ß	c3 9f	LATIN SMALL LETTER SHARP S
U+00E0	à	c3 a0	LATIN SMALL LETTER A WITH GRAVE
U+00E1	á	c3 a1	LATIN SMALL LETTER A WITH ACUTE
U+00E2	â	c3 a2	LATIN SMALL LETTER A WITH CIRCUMFLEX
U+00E3	ã	c3 a3	LATIN SMALL LETTER A WITH TILDE
U+00E4	ä	c3 a4	LATIN SMALL LETTER A WITH DIAERESIS

U+00E5	å	c3 a5	LATIN SMALL LETTER A WITH RING ABOVE
U+00E6	æ	c3 a6	LATIN SMALL LETTER AE
U+00E7	ç	c3 a7	LATIN SMALL LETTER C WITH CEDILLA
U+00E8	è	c3 a8	LATIN SMALL LETTER E WITH GRAVE
U+00E9	é	c3 a9	LATIN SMALL LETTER E WITH ACUTE
U+00EA	ê	c3 aa	LATIN SMALL LETTER E WITH CIRCUMFLEX
U+00EB	ë	c3 ab	LATIN SMALL LETTER E WITH DIAERESIS
U+00EC	ì	c3 ac	LATIN SMALL LETTER I WITH GRAVE
U+00ED	í	c3 ad	LATIN SMALL LETTER I WITH ACUTE
U+00EE	î	c3 ae	LATIN SMALL LETTER I WITH CIRCUMFLEX
U+00EF	ï	c3 af	LATIN SMALL LETTER I WITH DIAERESIS
U+00F0	ð	c3 b0	LATIN SMALL LETTER ETH
U+00F1	ñ	c3 b1	LATIN SMALL LETTER N WITH TILDE
U+00F2	ò	c3 b2	LATIN SMALL LETTER O WITH GRAVE
U+00F3	ó	c3 b3	LATIN SMALL LETTER O WITH ACUTE
U+00F4	ô	c3 b4	LATIN SMALL LETTER O WITH CIRCUMFLEX
U+00F5	õ	c3 b5	LATIN SMALL LETTER O WITH TILDE
U+00F6	ö	c3 b6	LATIN SMALL LETTER O WITH DIAERESIS
U+00F7	÷	c3 b7	DIVISION SIGN
U+00F8	ø	c3 b8	LATIN SMALL LETTER O WITH STROKE
U+00F9	ù	c3 b9	LATIN SMALL LETTER U WITH GRAVE
U+00FA	ú	c3 ba	LATIN SMALL LETTER U WITH ACUTE
U+00FB	û	c3 bb	LATIN SMALL LETTER U WITH CIRCUMFLEX
U+00FC	ü	c3 bc	LATIN SMALL LETTER U WITH DIAERESIS
U+00FD	ý	c3 bd	LATIN SMALL LETTER Y WITH ACUTE
U+00FE	þ	c3 be	LATIN SMALL LETTER THORN
U+00FF	ÿ	c3 bf	LATIN SMALL LETTER Y WITH DIAERESIS